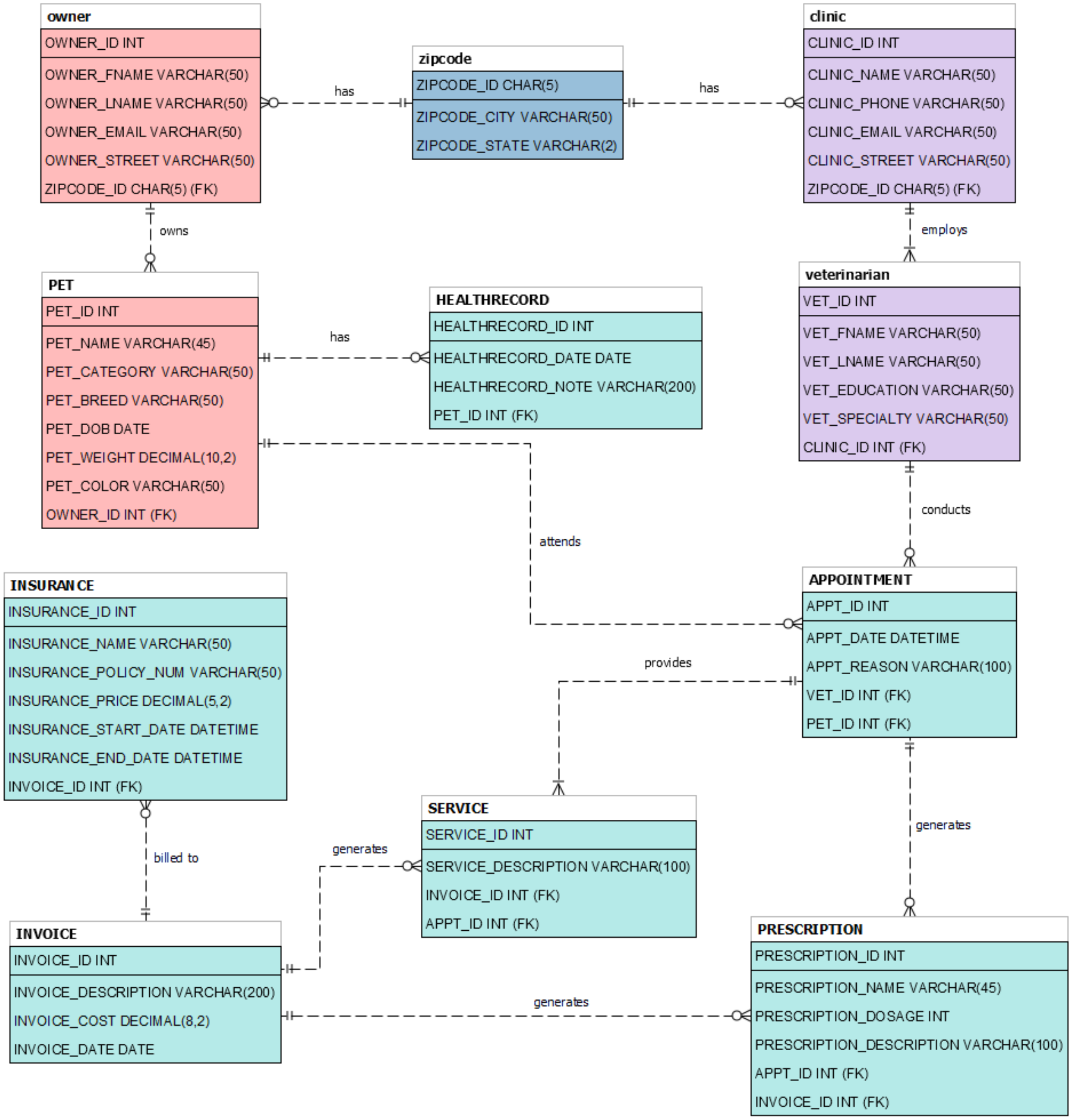
Grace Li
Michi Tsuji
Pete Mettaprasert
Nick McNamara
Group 4
Milestone 2
May 17th, 2022
CPSC 5021 Section 1

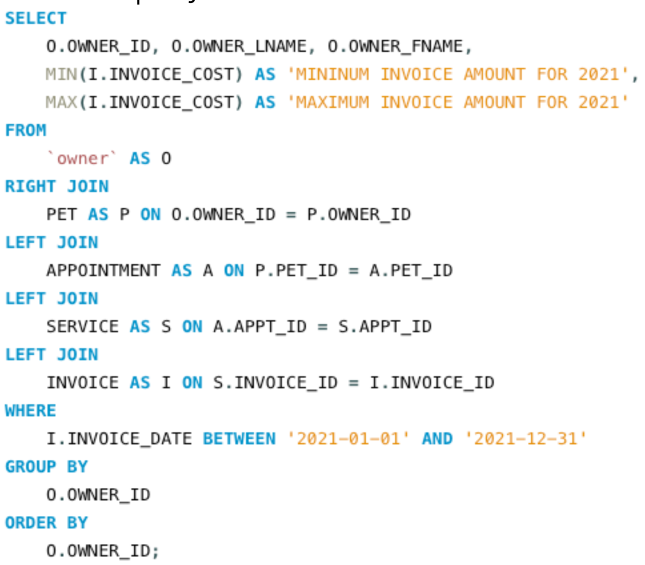
Internal Model:



Query 1:

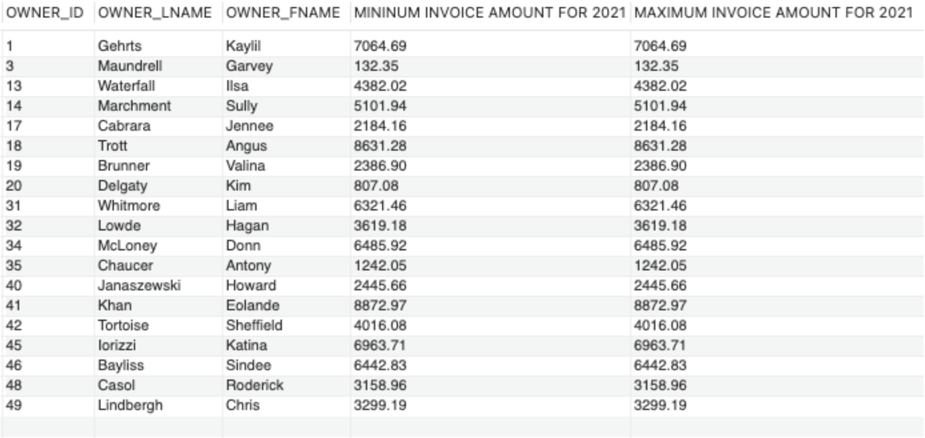
User(s): Clinic

This query shows the max and min invoice costs for each owner for calendar year 2021. Clinics can use this to see the range of invoice costs for each owner during the year. Currently, the min and max amounts are the same because our database has one invoice per owner, which isn't ideal to show the data we are trying to extract. But this is a useful query as data becomes more abundant.



```
SELECT
    O.OWNER_ID, O.OWNER_LNAME, O.OWNER_FNAME,
    MIN(I.INVOICE_COST) AS 'MININUM INVOICE AMOUNT FOR 2021',
    MAX(I.INVOICE_COST) AS 'MAXIMUM INVOICE AMOUNT FOR 2021'
FROM
    `owner` AS O
RIGHT JOIN
    PET AS P ON O.OWNER_ID = P.OWNER_ID
LEFT JOIN
    APPOINTMENT AS A ON P.PET_ID = A.PET_ID
LEFT JOIN
    SERVICE AS S ON A.APPT_ID = S.APPT_ID
LEFT JOIN
    INVOICE AS I ON S.INVOICE_ID = I.INVOICE_ID
WHERE
    I.INVOICE_DATE BETWEEN '2021-01-01' AND '2021-12-31'
GROUP BY
    O.OWNER_ID
ORDER BY
    O.OWNER_ID;
```

Query 1 Output:

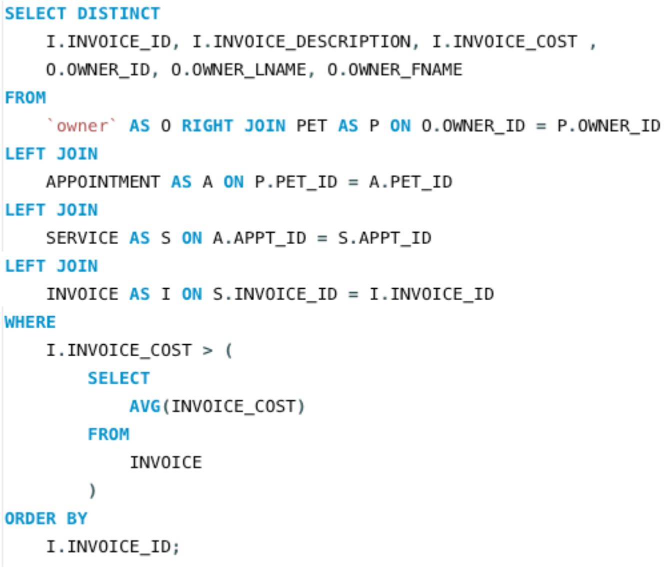


OWNER_ID	OWNER_LNAME	OWNER_FNAME	MININUM INVOICE AMOUNT FOR 2021	MAXIMUM INVOICE AMOUNT FOR 2021
1	Gehrts	Kaylil	7064.69	7064.69
3	Maundrell	Garvey	132.35	132.35
13	Waterfall	Ilsa	4382.02	4382.02
14	Marchment	Sully	5101.94	5101.94
17	Cabrara	Jennee	2184.16	2184.16
18	Trott	Angus	8631.28	8631.28
19	Brunner	Valina	2386.90	2386.90
20	Delgaty	Kim	807.08	807.08
31	Whitmore	Liam	6321.46	6321.46
32	Lowde	Hagan	3619.18	3619.18
34	McLoney	Donn	6485.92	6485.92
35	Chaucer	Antony	1242.05	1242.05
40	Janaszewski	Howard	2445.66	2445.66
41	Khan	Eolande	8872.97	8872.97
42	Tortoise	Sheffield	4016.08	4016.08
45	Iorizzi	Katina	6963.71	6963.71
46	Bayliss	Sindee	6442.83	6442.83
48	Casol	Roderick	3158.96	3158.96
49	Lindbergh	Chris	3299.19	3299.19

Query 2:

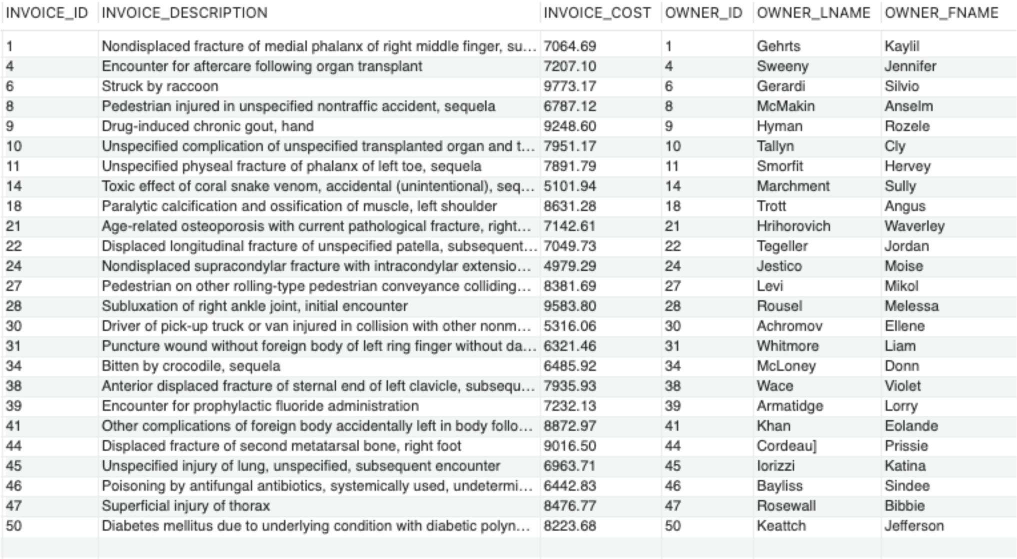
User(s): Clinic

This query shows invoice costs larger than the average invoice cost and their associated owners. Clinics can use this information to determine the higher cost services in the market and the larger clients. They can then compare this information to structure their own operations and pricing.



```
SELECT DISTINCT
    I.INVOICE_ID, I.INVOICE_DESCRIPTION, I.INVOICE_COST ,
    O.OWNER_ID, O.OWNER_LNAME, O.OWNER_FNAME
FROM
    `owner` AS O RIGHT JOIN PET AS P ON O.OWNER_ID = P.OWNER_ID
LEFT JOIN
    APPOINTMENT AS A ON P.PET_ID = A.PET_ID
LEFT JOIN
    SERVICE AS S ON A.APPT_ID = S.APPT_ID
LEFT JOIN
    INVOICE AS I ON S.INVOICE_ID = I.INVOICE_ID
WHERE
    I.INVOICE_COST > (
        SELECT
            AVG(INVOICE_COST)
        FROM
            INVOICE
        )
ORDER BY
    I.INVOICE_ID;
```

Query 2 Output:



INVOICE_ID	INVOICE_DESCRIPTION	INVOICE_COST	OWNER_ID	OWNER_LNAME	OWNER_FNAME
1	Nondisplaced fracture of medial phalanx of right middle finger, su...	7064.69	1	Gehrts	Kaylil
4	Encounter for aftercare following organ transplant	7207.10	4	Sweeny	Jennifer
6	Struck by raccoon	9773.17	6	Gerardi	Silvio
8	Pedestrian injured in unspecified nontraffic accident, sequela	6787.12	8	McMakin	Anselm
9	Drug-induced chronic gout, hand	9248.60	9	Hyman	Rozele
10	Unspecified complication of unspecified transplanted organ and t...	7951.17	10	Tallyn	Cly
11	Unspecified physeal fracture of phalanx of left toe, sequela	7891.79	11	Smorfit	Hervey
14	Toxic effect of coral snake venom, accidental (unintentional), seq...	5101.94	14	Marchment	Sully
18	Paralytic calcification and ossification of muscle, left shoulder	8631.28	18	Trott	Angus
21	Age-related osteoporosis with current pathological fracture, right...	7142.61	21	Hrihorovich	Waverley
22	Displaced longitudinal fracture of unspecified patella, subsequent...	7049.73	22	Tegeller	Jordan
24	Nondisplaced supracondylar fracture with intracondylar extensio...	4979.29	24	Jestico	Moise
27	Pedestrian on other rolling-type pedestrian conveyance colliding...	8381.69	27	Levi	Mikol
28	Subluxation of right ankle joint, initial encounter	9583.80	28	Rousel	Melessa
30	Driver of pick-up truck or van injured in collision with other nonm...	5316.06	30	Achromov	Ellene
31	Puncture wound without foreign body of left ring finger without da...	6321.46	31	Whitmore	Liam
34	Bitten by crocodile, sequela	6485.92	34	McLoney	Donn
38	Anterior displaced fracture of sternal end of left clavicle, subsequ...	7935.93	38	Wace	Violet
39	Encounter for prophylactic fluoride administration	7232.13	39	Armatidge	Lorry
41	Other complications of foreign body accidentally left in body follo...	8872.97	41	Khan	Eolande
44	Displaced fracture of second metatarsal bone, right foot	9016.50	44	Cordeau]	Prissie
45	Unspecified injury of lung, unspecified, subsequent encounter	6963.71	45	Iorizzi	Katina
46	Poisoning by antifungal antibiotics, systemically used, undetermi...	6442.83	46	Bayliss	Sindee
47	Superficial injury of thorax	8476.77	47	Rosewall	Bibbie
50	Diabetes mellitus due to underlying condition with diabetic polyn...	8223.68	50	Keattch	Jefferson

Query 3:

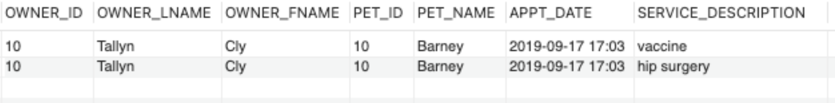
User(s): Vet, Owner

This query pulls up the pet's vaccination dates based on the owner last name or their pet's name. Both the veterinarian and the owner can use this information to see a pet's vaccination history and determine whether the pet is up to date with their vaccination.

The current patient has had only one vaccination, hence, one record date.

```
SELECT
    O.OWNER_ID, O.OWNER_LNAME, O.OWNER_FNAME,
    P.PET_ID, P.PET_NAME,
    A.APPT_DATE,
    S.SERVICE_DESCRIPTION
FROM
    `owner` AS O
RIGHT JOIN
    PET AS P ON O.OWNER_ID = P.OWNER_ID
LEFT JOIN
    APPOINTMENT AS A ON P.PET_ID = A.PET_ID
LEFT JOIN
    SERVICE AS S ON A.APPT_ID = S.APPT_ID
WHERE
    S.SERVICE_DESCRIPTION = 'VACCINE'
    AND
        O.OWNER_LNAME = 'TALLYN' OR P.PET_NAME = 'BARNEY'
ORDER BY
    O.OWNER_ID, P.PET_ID;
```

Query 3 Output:



OWNER_ID	OWNER_LNAME	OWNER_FNAME	PET_ID	PET_NAME	APPT_DATE	SERVICE_DESCRIPTION
10	Tallyn	Cly	10	Barney	2019-09-17 17:03	vaccine
10	Tallyn	Cly	10	Barney	2019-09-17 17:03	hip surgery

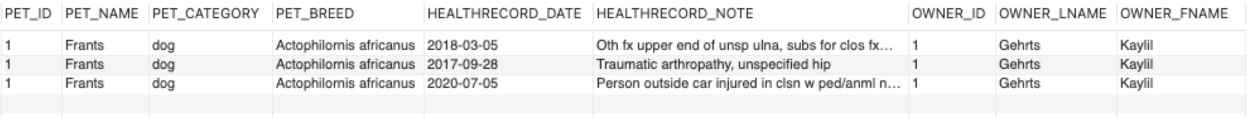
Query 4:

User(s): Vet

This query lists out a pet's basic information, health record, and their owner. At each appointment, the veterinarian can use this information to understand the patient's history and to determine how to proceed with treatments. The association of the pet's name to its owner's name also adds an extra identification verification step.

```
SELECT
    P.PET_ID, P.PET_NAME, P.PET_CATEGORY, P.PET_BREED,
    H.HEALTHRECORD_DATE, H.HEALTHRECORD_NOTE,
    P.OWNER_ID,
    O.OWNER_LNAME, O.OWNER_FNAME
FROM
    PET P
LEFT JOIN
    `owner` O ON O.OWNER_ID = P.OWNER_ID
LEFT JOIN
    HEALTHRECORD H ON P.PET_ID = H.PET_ID
WHERE
    P.PET_NAME = 'FRANTS';
```

Query 4 Output:

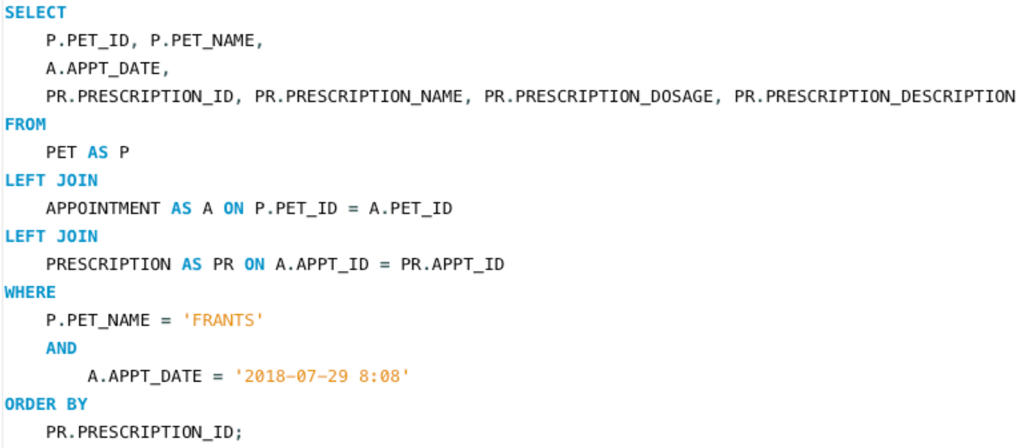


PET_ID	PET_NAME	PET_CATEGORY	PET_BREED	HEALTHRECORD_DATE	HEALTHRECORD_NOTE	OWNER_ID	OWNER_LNAME	OWNER_FNAME
1	Frants	dog	Actophilornis africanus	2018-03-05	Oth fx upper end of unsp ulna, subs for clos fx…	1	Gehrts	Kaylil
1	Frants	dog	Actophilornis africanus	2017-09-28	Traumatic arthropathy, unspecified hip	1	Gehrts	Kaylil
1	Frants	dog	Actophilornis africanus	2020-07-05	Person outside car injured in clsn w ped/anml n…	1	Gehrts	Kaylil

Query 5:

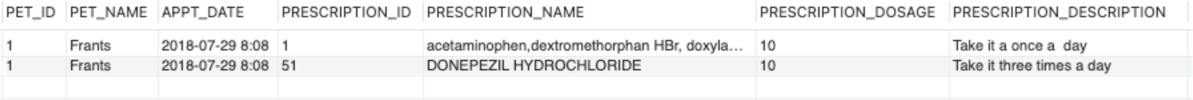
User(s): Owner

This query lists prescription information for specific pet on a specific appointment date. This provides the owner a summary of their pet's medication and instructions after the appointment visit.



```
SELECT
    P.PET_ID, P.PET_NAME,
    A.APPT_DATE,
    PR.PRESCRIPTION_ID, PR.PRESCRIPTION_NAME, PR.PRESCRIPTION_DOSAGE, PR.PRESCRIPTION_DESCRIPTION
FROM
    PET AS P
LEFT JOIN
    APPOINTMENT AS A ON P.PET_ID = A.PET_ID
LEFT JOIN
    PRESCRIPTION AS PR ON A.APPT_ID = PR.APPT_ID
WHERE
    P.PET_NAME = 'FRANTS'
    AND
        A.APPT_DATE = '2018-07-29 8:08'
ORDER BY
    PR.PRESCRIPTION_ID;
```

Query 5 Output:

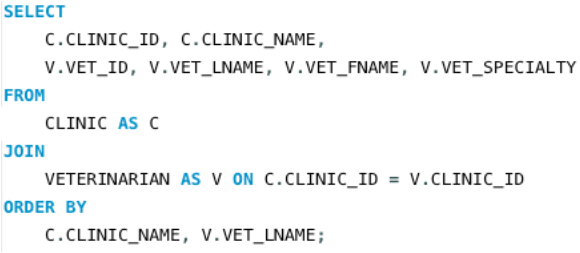


PET_ID	PET_NAME	APPT_DATE	PRESCRIPTION_ID	PRESCRIPTION_NAME	PRESCRIPTION_DOSAGE	PRESCRIPTION_DESCRIPTION
1	Frants	2018-07-29 8:08	1	acetaminophen,dextromethorphan HBr, doxyla...	10	Take it a once a day
1	Frants	2018-07-29 8:08	51	DONEPEZIL HYDROCHLORIDE	10	Take it three times a day

Query 6:

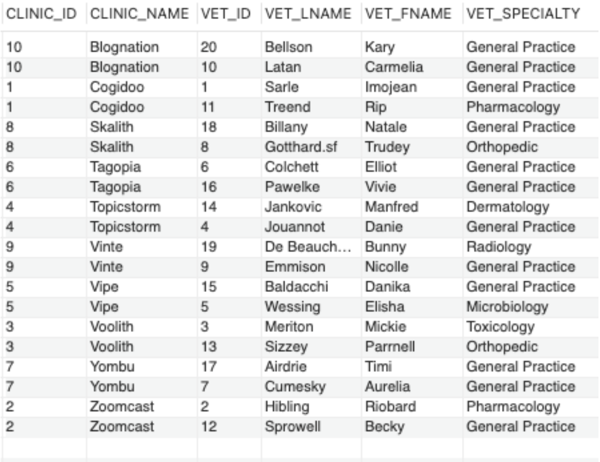
User(s): Owner, Clinic

This query lists the veterinarians and their specialties for each clinic, sorted by clinic name and veterinarians' last names. Owners can use this to find general or specialized veterinarians for their pets at each clinic. Clinics can use this to view its current employees and manage its future employment planning.



```
SELECT
    C.CLINIC_ID, C.CLINIC_NAME,
    V.VET_ID, V.VET_LNAME, V.VET_FNAME, V.VET_SPECIALTY
FROM
    CLINIC AS C
JOIN
    VETERINARIAN AS V ON C.CLINIC_ID = V.CLINIC_ID
ORDER BY
    C.CLINIC_NAME, V.VET_LNAME;
```

Query 6 Output:



CLINIC_ID	CLINIC_NAME	VET_ID	VET_LNAME	VET_FNAME	VET_SPECIALTY
10	Blognation	20	Bellson	Kary	General Practice
10	Blognation	10	Latan	Carmelia	General Practice
1	Cogidoo	1	Sarle	Imojean	General Practice
1	Cogidoo	11	Treend	Rip	Pharmacology
8	Skalith	18	Billany	Natale	General Practice
8	Skalith	8	Gotthard.sf	Trudey	Orthopedic
6	Tagopia	6	Colchett	Elliot	General Practice
6	Tagopia	16	Pawelke	Vivie	General Practice
4	Topicstorm	14	Jankovic	Manfred	Dermatology
4	Topicstorm	4	Jouannot	Danie	General Practice
9	Vinte	19	De Beauch…	Bunny	Radiology
9	Vinte	9	Emmison	Nicolle	General Practice
5	Vipe	15	Baldacchi	Danika	General Practice
5	Vipe	5	Wessing	Elisha	Microbiology
3	Voolith	3	Meriton	Mickie	Toxicology
3	Voolith	13	Sizzey	Parrnell	Orthopedic
7	Yombu	17	Airdrie	Timi	General Practice
7	Yombu	7	Cumesky	Aurelia	General Practice
2	Zoomcast	2	Hibling	Riobard	Pharmacology
2	Zoomcast	12	Sprowell	Becky	General Practice

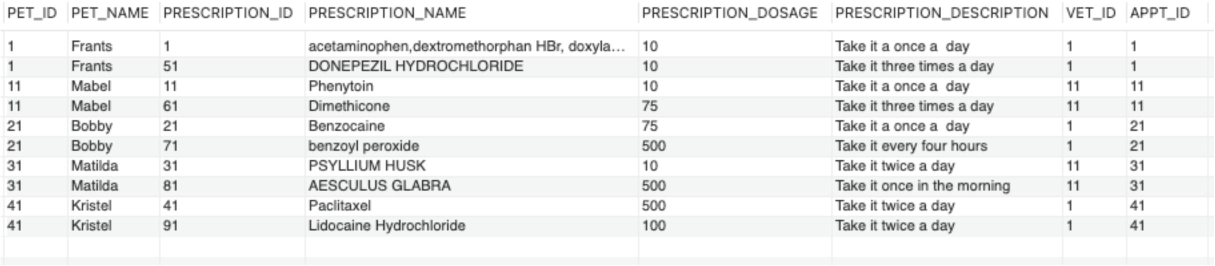
<u>Query 7</u>:

User(s): Clinic

This query returns the names of all pets and their prescriptions, prescribed by providers at the clinic Cogidoo. The results include detailed information about each prescription including dosage and instructions. The output of this query can be used by Cogidoo administrators for supply chain planning, as they can see all possible medications being used by their patients, and therefore which medications should be kept in stock at the clinic.

```
SELECT
    P.PET_ID, P.PET_NAME,
    RX.PRESCRIPTION_ID, RX.PRESCRIPTION_NAME, RX.PRESCRIPTION_DOSAGE, RX.PRESCRIPTION_DESCRIPTION,
    A.VET_ID, RX.APPT_ID
FROM
    pet P
INNER JOIN
    appointment A ON P.PET_ID
INNER JOIN
    prescription RX ON RX.APPT_ID
WHERE
    A.PET_ID = P.PET_ID
    AND
        RX.APPT_ID = A.APPT_ID
        AND
            A.VET_ID IN (
                SELECT
                    V.VET_ID
                FROM
                    veterinarian AS V
                WHERE
                    V.CLINIC_ID = (
                        SELECT
                            C.CLINIC_ID
                        FROM
                            clinic AS C
                        WHERE
                            C.CLINIC_NAME LIKE '%cogidoo%'
                        )
            )
ORDER BY
    P.PET_ID,
    RX.PRESCRIPTION_ID;
```

Query 7 Output:

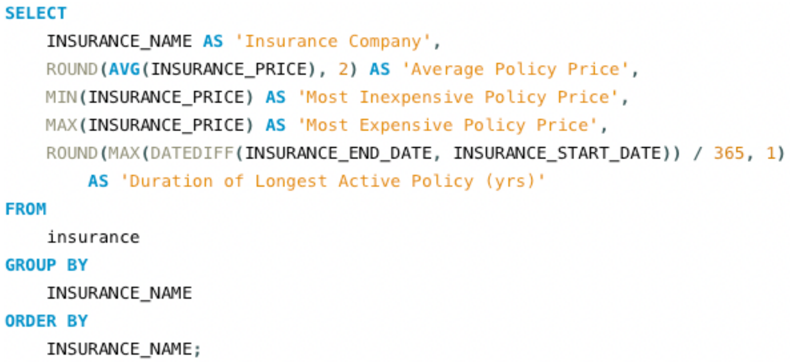


PET_ID	PET_NAME	PRESCRIPTION_ID	PRESCRIPTION_NAME	PRESCRIPTION_DOSAGE	PRESCRIPTION_DESCRIPTION	VET_ID	APPT_ID
1	Frants	1	acetaminophen,dextromethorphan HBr, doxyla...	10	Take it a once a day	1	1
1	Frants	51	DONEPEZIL HYDROCHLORIDE	10	Take it three times a day	1	1
11	Mabel	11	Phenytoin	10	Take it a once a day	11	11
11	Mabel	61	Dimethicone	75	Take it three times a day	11	11
21	Bobby	21	Benzocaine	75	Take it a once a day	1	21
21	Bobby	71	benzoyl peroxide	500	Take it every four hours	1	21
31	Matilda	31	PSYLLIUM HUSK	10	Take it twice a day	11	31
31	Matilda	81	AESCULUS GLABRA	500	Take it once in the morning	11	31
41	Kristel	41	Paclitaxel	500	Take it twice a day	1	41
41	Kristel	91	Lidocaine Hydrochloride	100	Take it twice a day	1	41

Query 8:

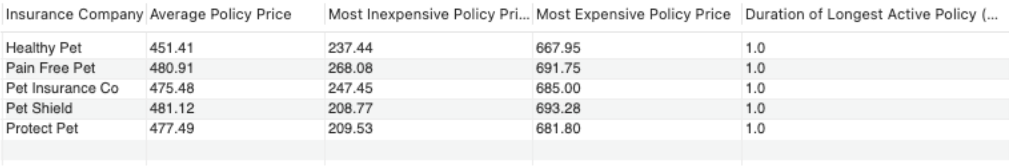
User(s): Owner

For each pet insurance company, this query returns the average policy price, the most inexpensive price for a policy, the most expensive price for a policy, the duration of the longest active policy. This query can be used to generate a report for pet owners that are browsing for a new insurance policy, as the data returned can be used in shopping decision-making.



```
SELECT
    INSURANCE_NAME AS 'Insurance Company',
    ROUND(AVG(INSURANCE_PRICE), 2) AS 'Average Policy Price',
    MIN(INSURANCE_PRICE) AS 'Most Inexpensive Policy Price',
    MAX(INSURANCE_PRICE) AS 'Most Expensive Policy Price',
    ROUND(MAX(DATEDIFF(INSURANCE_END_DATE, INSURANCE_START_DATE)) / 365, 1)
        AS 'Duration of Longest Active Policy (yrs)'
FROM
    insurance
GROUP BY
    INSURANCE_NAME
ORDER BY
    INSURANCE_NAME;
```

Query 8 Output:



Insurance Company	Average Policy Price	Most Inexpensive Policy Pri...	Most Expensive Policy Price	Duration of Longest Active Policy (...
Healthy Pet	451.41	237.44	667.95	1.0
Pain Free Pet	480.91	268.08	691.75	1.0
Pet Insurance Co	475.48	247.45	685.00	1.0
Pet Shield	481.12	208.77	693.28	1.0
Protect Pet	477.49	209.53	681.80	1.0

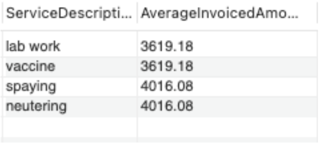
Query 9:

User(s): Clinic

This query returns the name of the service and average service cost for all services provided in 2021 by the Zoomcast clinic. The output of the query can be used in a report designed for the Zoomcast clinic, where they can do a lookback on prior year services and the average amount invoiced for the services to aid in business decisions and assess their most premium service.

```
SELECT
    S.SERVICE_DESCRIPTION AS ServiceDescription,
    ROUND(AVG(I.INVOICE_COST), 2) AS AverageInvoicedAmount
FROM
    service S,
    invoice I,
    prescription RX,
    appointment A,
    veterinarian V,
    clinic C
WHERE
    S.INVOICE_ID = I.INVOICE_ID
    AND
        I.INVOICE_ID = RX.INVOICE_ID
        AND
            RX.APPT_ID = A.APPT_ID
            AND
                A.VET_ID = V.VET_ID
                AND
                    V.CLINIC_ID = C.CLINIC_ID
                    AND
                        C.CLINIC_NAME LIKE '%zoomcast%'
                        AND
                            I.INVOICE_DATE BETWEEN '2021-01-01' AND '2021-12-31'
GROUP BY
    S.SERVICE_DESCRIPTION
ORDER BY
    AverageInvoicedAmount ASC;
```

Query 9 Output:



ServiceDescripti...	AverageInvoicedAmo...
lab work	3619.18
vaccine	3619.18
spaying	4016.08
neutering	4016.08

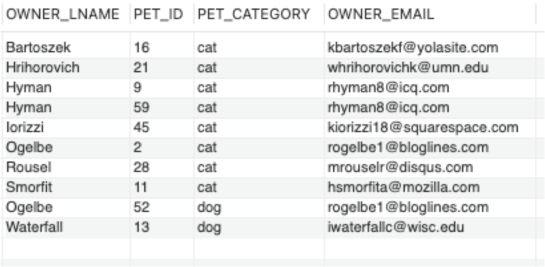
Query 10:

User(s): Clinic

This query returns the owner information for owners in the database who own a dog or a cat and visited the Vipe clinic. This can help the clinic track their clientele needs and what kinds of vets they need to hire to service their clients.

```
SELECT DISTINCT
    O.OWNER_LNAME, P.PET_ID, P.PET_CATEGORY, O.OWNER_EMAIL
FROM
    pet AS P,
    `owner` AS O,
    clinic AS C,
    zipcode AS Z
WHERE
    P.PET_CATEGORY IN ('cat', 'dog')
    AND
        O.OWNER_ID = P.OWNER_ID
        AND
            C.ZIPCODE_ID = Z.ZIPCODE_ID
            AND
                O.ZIPCODE_ID = Z.ZIPCODE_ID
                AND C.CLINIC_NAME LIKE 'vipe'
ORDER BY
    P.PET_CATEGORY,
    O.OWNER_LNAME,
    P.PET_ID;
```

Query 10 Output:



OWNER_LNAME	PET_ID	PET_CATEGORY	OWNER_EMAIL
Bartoszek	16	cat	kbartoszekf@yolasite.com
Hrihorovich	21	cat	whrihorovichk@umn.edu
Hyman	9	cat	rhyman8@icq.com
Hyman	59	cat	rhyman8@icq.com
Iorizzi	45	cat	kiorizzi18@squarespace.com
Ogelbe	2	cat	rogelbe1@bloglines.com
Rousel	28	cat	mrouselr@disqus.com
Smorfit	11	cat	hsmorfita@mozilla.com
Ogelbe	52	dog	rogelbe1@bloglines.com
Waterfall	13	dog	iwaterfallc@wisc.edu

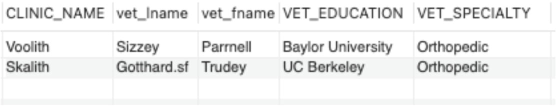
Query 11:

User(s): Owner

This query allows a pet owner to find which doctors specialize in a specific specialty, in this case orthopedic work. This is useful to the pet's owner and the clinic system because it quickly allows clients to look up clinics that will service their needs, and will help vets only see patients they can actually treat.

```
SELECT
    C.CLINIC_NAME,
    V.vet_lname,
    V.vet_fname,
    V.VET_EDUCATION,
    V.VET_SPECIALTY
FROM
    clinic as C,
    veterinarian as V
WHERE
    V.VET_SPECIALTY like 'orthopedic'
    AND
        C.CLINIC_ID = V.CLINIC_ID
ORDER BY
    C.CLINIC_NAME DESC,
    V.VET_LNAME;
```

Query 11 Output:

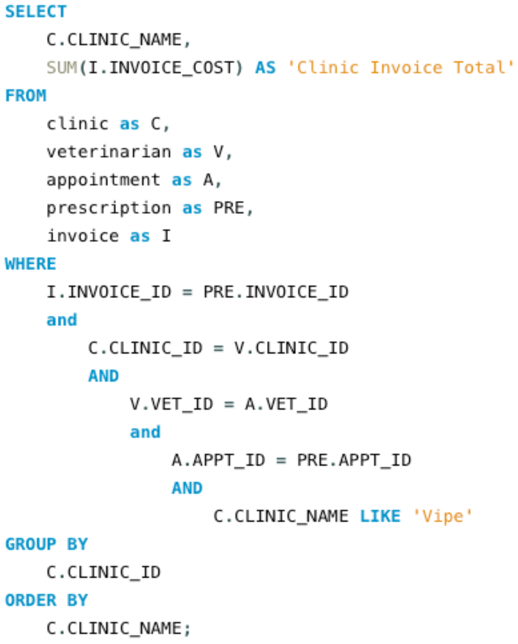


CLINIC_NAME	vet_lname	vet_fname	VET_EDUCATION	VET_SPECIALTY
Voolith	Sizzey	Parrnell	Baylor University	Orthopedic
Skalith	Gotthard.sf	Trudey	UC Berkeley	Orthopedic

Query 12:

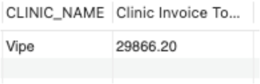
User(s): Clinic

This query returns the sum of the invoices processed by the Vipe clinic. This allows the clinics to see how much revenue was generated through invoices and can help in tracking revenue generated by each clinic.



```
SELECT
    C.CLINIC_NAME,
    SUM(I.INVOICE_COST) AS 'Clinic Invoice Total'
FROM
    clinic as C,
    veterinarian as V,
    appointment as A,
    prescription as PRE,
    invoice as I
WHERE
    I.INVOICE_ID = PRE.INVOICE_ID
    and
        C.CLINIC_ID = V.CLINIC_ID
        AND
            V.VET_ID = A.VET_ID
            and
                A.APPT_ID = PRE.APPT_ID
                AND
                    C.CLINIC_NAME LIKE 'Vipe'
GROUP BY
    C.CLINIC_ID
ORDER BY
    C.CLINIC_NAME;
```

Query 12 Output:



CLINIC_NAME	Clinic Invoice To...
Vipe	29866.20

Stored Procedure 1:

User(s): Clinic

This stored procedure produces a report that shows individual invoice details, labeled with their aging buckets based on the date of the invoice – 0-30 days, 31-60 days, etc. This is important for the clinic's Accounts Receivable/Collections department to see which invoices are significantly aged, so they can reach out to invoice owners to collect payments.

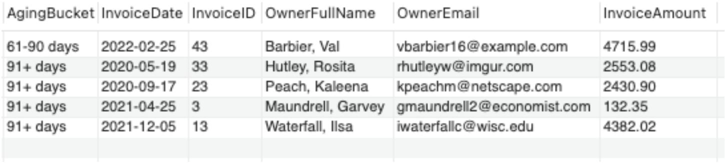
```
DELIMITER $$
CREATE DEFINER=`root`@`localhost`
PROCEDURE `clinicInvoiceAgingDetail`(
    IN the_clinic VARCHAR(50)
)
    READS SQL DATA
BEGIN
    SELECT DISTINCT
    CASE
        WHEN DATEDIFF(CURRENT_DATE(), I.INVOICE_DATE) < 31 THEN '0-30 days'
        WHEN DATEDIFF(CURRENT_DATE(), I.INVOICE_DATE) > 30 AND DATEDIFF(CURRENT_DATE(), I.INVOICE_DATE) < 61 THEN '31-60 days'
        WHEN DATEDIFF(CURRENT_DATE(), I.INVOICE_DATE) > 60 AND DATEDIFF(CURRENT_DATE(), I.INVOICE_DATE) < 91 THEN '61-90 days'
        WHEN DATEDIFF(CURRENT_DATE(), I.INVOICE_DATE) > 90 THEN '91+ days'
        ELSE 'No aging'
    END AS AgingBucket,
    I.INVOICE_DATE AS InvoiceDate,
    I.INVOICE_ID AS InvoiceID,
    CONCAT(O.OWNER_LNAME, ', ', O.OWNER_FNAME) AS 'OwnerFullName',
    O.OWNER_EMAIL AS OwnerEmail,
    I.INVOICE_COST AS InvoiceAmount
    FROM
        clinic C
    LEFT JOIN
        veterinarian V ON C.CLINIC_ID = V.CLINIC_ID
    LEFT JOIN
        appointment A ON V.VET_ID = A.VET_ID
    LEFT JOIN
        prescription RX ON A.APPT_ID = RX.APPT_ID
    LEFT JOIN
        invoice I ON RX.INVOICE_ID = I.INVOICE_ID
    LEFT JOIN
        pet P ON A.PET_ID = P.PET_ID
    LEFT JOIN
        `owner` O ON P.OWNER_ID = O.OWNER_ID
    WHERE
        C.CLINIC_NAME = the_clinic
ORDER BY
    AgingBucket ASC, InvoiceDate ASC;
END$$
DELIMITER ;
```

Stored Procedure 1 – example calls:

```
CALL clinicInvoiceAgingDetail('voolith');
CALL clinicInvoiceAgingDetail('vipe');
```

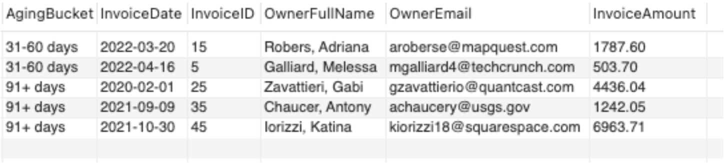
Stored Procedure 1 Output:

Voolith Clinic:



AgingBucket	InvoiceDate	InvoiceID	OwnerFullName	OwnerEmail	InvoiceAmount
61-90 days	2022-02-25	43	Barbier, Val	vbarbier16@example.com	4715.99
91+ days	2020-05-19	33	Hutley, Rosita	rhutleyw@imgur.com	2553.08
91+ days	2020-09-17	23	Peach, Kaleena	kpeachm@netscape.com	2430.90
91+ days	2021-04-25	3	Maundrell, Garvey	gmaundrell2@economist.com	132.35
91+ days	2021-12-05	13	Waterfall, Ilsa	iwaterfallc@wisc.edu	4382.02

Vipe Clinic:



AgingBucket	InvoiceDate	InvoiceID	OwnerFullName	OwnerEmail	InvoiceAmount
31-60 days	2022-03-20	15	Robers, Adriana	aroberse@mapquest.com	1787.60
31-60 days	2022-04-16	5	Galliard, Melessa	mgalliard4@techcrunch.com	503.70
91+ days	2020-02-01	25	Zavattieri, Gabi	gzavattierio@quantcast.com	4436.04
91+ days	2021-09-09	35	Chaucer, Antony	achaucery@usgs.gov	1242.05
91+ days	2021-10-30	45	Iorizzi, Katina	kiorizzi18@squarespace.com	6963.71

Stored Procedure 2:

User(s): Clinic

This stored procedure summarizes the information from stored procedure 1, and produces a report that shows the total for each aging bucket. This is important for the clinic as they can take a top-down look into their receivables and assess how the clinic as a business is performing, and whether they project to have enough cash on hand for day-to-day operations.

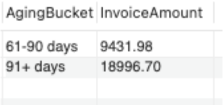
```
DELIMITER $$
CREATE DEFINER=`root`@`localhost`
PROCEDURE `clinicInvoiceAgingSummary`(
    IN the_clinic VARCHAR(50)
)
    READS SQL DATA
BEGIN
    SELECT DISTINCT
    CASE
        WHEN DATEDIFF(CURRENT_DATE(), I.INVOICE_DATE) < 31 THEN '0-30 days'
        WHEN DATEDIFF(CURRENT_DATE(), I.INVOICE_DATE) > 30 AND DATEDIFF(CURRENT_DATE(), I.INVOICE_DATE) < 61 THEN '31-60 days'
        WHEN DATEDIFF(CURRENT_DATE(), I.INVOICE_DATE) > 60 AND DATEDIFF(CURRENT_DATE(), I.INVOICE_DATE) < 91 THEN '61-90 days'
        WHEN DATEDIFF(CURRENT_DATE(), I.INVOICE_DATE) > 90 THEN '91+ days'
        ELSE 'No aging'
    END AS AgingBucket,
    SUM(I.INVOICE_COST) AS InvoiceAmount
    FROM
        clinic C
    LEFT JOIN
        veterinarian V ON C.CLINIC_ID = V.CLINIC_ID
    LEFT JOIN
        appointment A ON V.VET_ID = A.VET_ID
    LEFT JOIN
        prescription RX ON A.APPT_ID = RX.APPT_ID
    LEFT JOIN
        invoice I ON RX.INVOICE_ID = I.INVOICE_ID
    WHERE
        C.CLINIC_NAME = the_clinic
    GROUP BY
        AgingBucket
    ORDER BY
        AgingBucket ASC;
END$$
DELIMITER ;
```

Stored Procedure 2 – example calls:

```
CALL clinicInvoiceAgingSummary('voolith');
CALL clinicInvoiceAgingSummary('vipe');
```

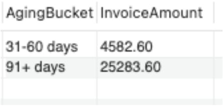
Stored Procedure 2 Output:

Voolith Clinic:



AgingBucket	InvoiceAmount
61-90 days	9431.98
91+ days	18996.70

Vipe Clinic:



AgingBucket	InvoiceAmount
31-60 days	4582.60
91+ days	25283.60